TUTORIAL 1

1 Remainder of a sparse polynomial

In this exercise we are interested in computing a remainder of a sparse polynomial S after dividing by a polynomial D, where $S, D \in K[X]$. (Assume that operations in K have unit cost.)

1. Give an example showing that assuming that S is sparse does not lead to better bounds for the classical division algorithm.
2. What is the cost of addition and multiplication in $K[X]/(D(X))$?
3. Show that one can compute $X^N \bmod D(X)$ in time $O((\deg D)^2 \log N)$. (Hint: use fast exponentiation.)
4. Assume that S has ω nonzero terms. Show that you get an algorithm of complexity $O(\omega(\deg D)^2 \log \deg S)$ which beats the classical division for ω at most $\frac{\deg S - \deg D}{\deg D \log \deg S}$.

2 Exact division

The goal of this exercise is to show that we can have a constant gain on (naive) polynomial division in the case where we know beforehand that the division is exact. The algorithm we are going to develop here is due to T. Jebelean.

Let $A(X) = \sum_{k=0}^{2n-1} a_k X^k$ and $B(X) = \sum_{k=0}^{n-1} b_k X^k$.

1. Prove that one can compute $Q_\ell = \sum_{k=n-\ell}^{n} q_k X^k$ such that $\deg(A - BQ_\ell) < 2n - 1 - \ell$ using $\ell + 1$ divisions and $\ell(\ell+1)/2$ multiplications in K. *Note that we are not interested in the remainder $A - BQ_\ell$, only in Q_ℓ.*
2. Prove that the algorithm of **1**. can be used to compute $S_\ell = \sum_{k=0}^{\ell} s_k X^k$ such that $\mathrm{val}(A - S_\ell B) > \ell$ using $\ell + 1$ divisions and $\ell(\ell+1)/2$ multiplications in K. We denote by $\mathrm{val}(P)$ the smallest degree of monomial of P with nonzero coefficient.
3. Assume that we know, for some reason, that $B|A$ and want to compute A/B. Use **1** & **2** to give an algorithm for this task, and compare this with the "schoolbook" division.
4. We only counted divisions and multiplications, but in a standard algebraic model, addition and subtraction also have a comparable cost. Does our result really make sense?

3 Multiplication of bivariate polynomials

In this exercise we analyze the complexity of Toom-Cook polynomial multiplication. As a side-result, we show how to efficiently multiply two bivariate polynomials. We again assume that operations in K have unit cost.

1. Let $A(X, Y)$ be a polynomial of degree at most D_1 in X and at most D_2 in Y. Give an algorithm that computes $A(X, c)$ for a given $c \in K$. What is its cost?
2. Let $R_1(X), \ldots, R_t(X)$ be polynomials of degree at most D_1, and $c_1, \ldots, c_t$ pairwise distinct elements in K. Give an algorithm of complexity $O(t^2(D_1 + 1))$ that returns $S(X, Y)$ of the form $\sum_{j=0}^{t-1} S_j(X) Y^j$, and such that $S(X, c_i) = R_i(X)$.

3. Give a full analysis of Toom-Cook's algorithm using the split $D_1 = n/r$ and $D_2 = r$.

4. What is the cost of a naive multiplication of two bivariate polynomials A and B of X-degree at most D_1 and Y-degree at most D_2?

5. Describe an evaluation-interpolation algorithm for multiplying bivariate polynomials.